

## Overview

In this lab, you will be introduced to technical debt, how to configure your Azure Build definitions to use SonarCloud, how to understand the analysis results, and finally how to configure quality profile to control the rule set used by SonarCloud for analyzing your project.

Technical debt is the set of problems in a development effort that make forward progress on customer value inefficient. Technical debt saps productivity by making code hard to understand, fragile, time-consuming to change, difficult to validate, and creates unplanned work that blocks progress. Technical debt saps an org's strength due to high costs in customer support, and, eventually, some combination of these issues creates a larger problem that someone runs into. Technical debt is insidious. It starts small and grows over time through rushed changes, lack of context and lack of discipline. It can materialize out of nowhere even for a project regarded as clean at some point in time, due to a change in project circumstances: prototype code may be promoted to serve as the basis for a feature; code produced for the U.S. market may be proposed for international, instantly creating debt related to localizability; technologies evolve, but the app doesn't keep up.

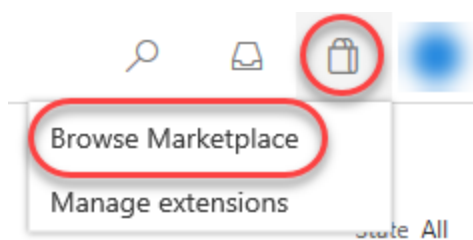
## Prerequisites

- A SonarCloud account from <https://sonarcloud.io>.
- This lab requires you to complete task 1 from the [prerequisite](#) instructions.

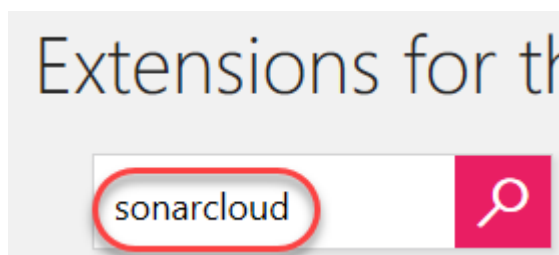
## Exercise 1: Managing Technical Debt with Azure DevOps and SonarCloud

### Task 1: Install and configure the SonarCloud extension

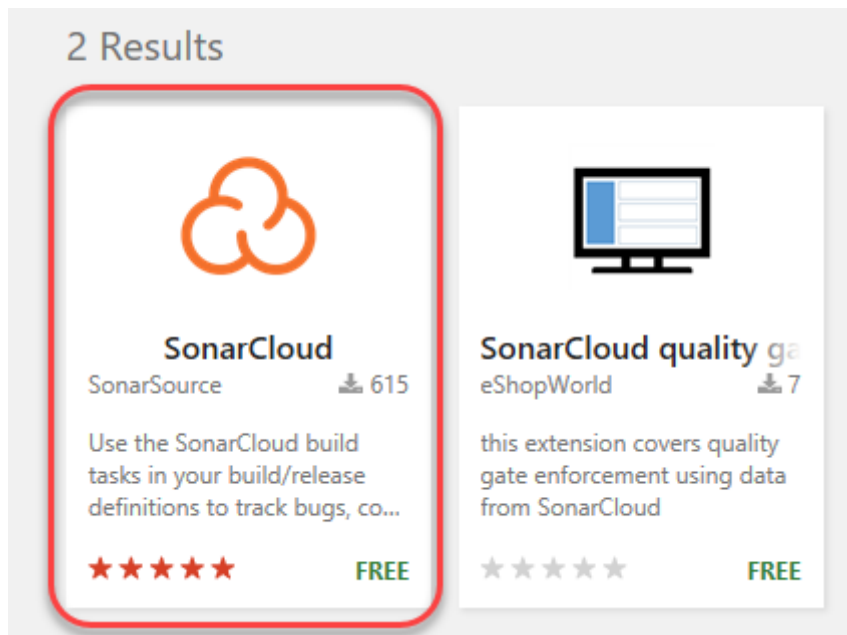
1. Navigate to your team project on Azure DevOps.
2. SonarCloud is provided as a marketplace extension. From the **Marketplace** navigation dropdown, select **Browse Marketplace**.



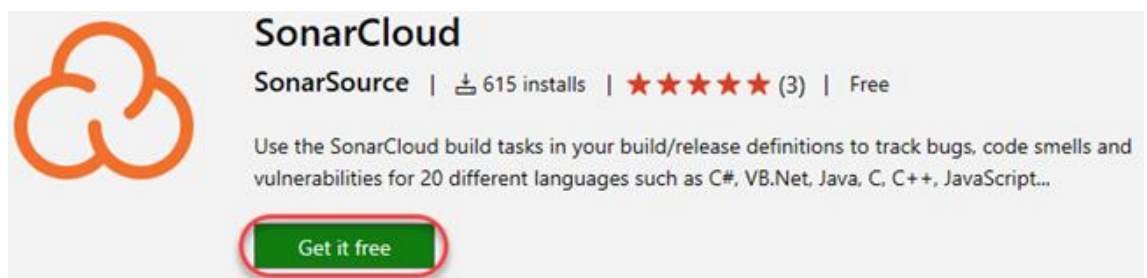
3. Search for "SonarCloud".



4. Select the **SonarCloud** option.



5. Click **Get it free**.



6. Select the organization to install **SonarCloud** into. This should be the organization that contains your **Parts Unlimited** project. Click **Install**.

Select a Visual Studio Team Services organization



7. Click **Proceed to organization**.

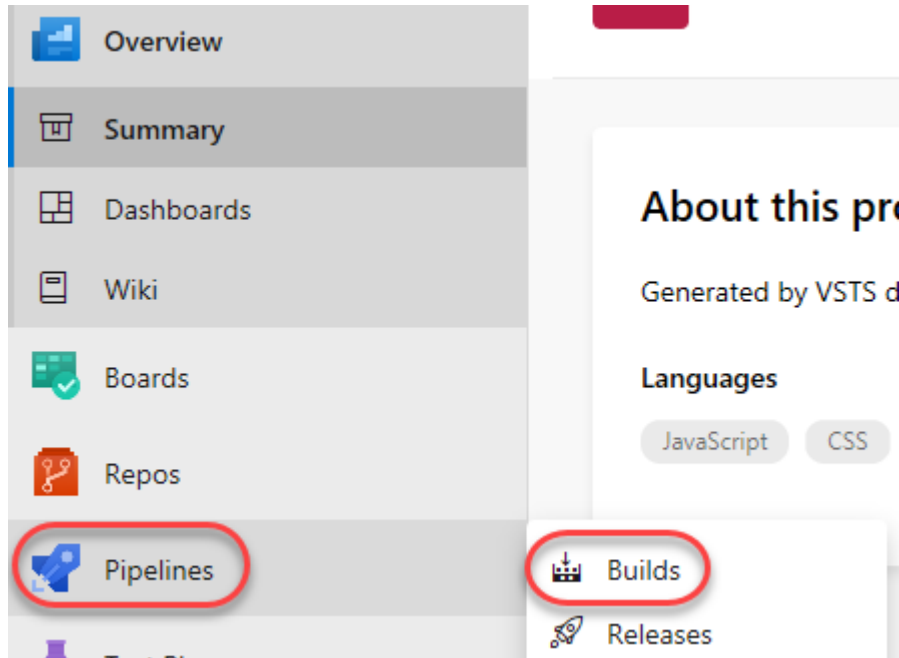
You are all set!



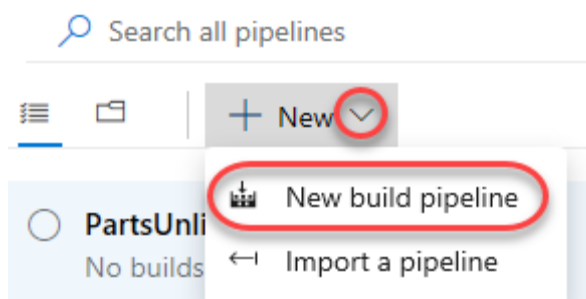
8. Navigate to your **Parts Unlimited** team project.

## **Task 2: Integrating a build with SonarCloud**

1. Navigate to **Pipelines | Builds**.



2. Select **New | New build pipeline** to create a new build pipeline.



3. Click **use the visual designer**.

## **Where is your code?**



**GitHub**

Home to the world's largest community of developers










**Azure Repos**

Free private Git repositories, pull requests, and code search



To create a pipeline for TFVC or Bitbucket Cloud, [use the visual designer.](#)

4. Accept the default options and click **Continue**.



Select a source

 Azure Repos Git	 TFVC	 GitHub	 GitHub Enterprise	 Subversion	 Bitbucket Cloud	 External Git
--	---	---	--	---	--	---



Team project

 Parts Unlimited Refresh 

Repository

 PartsUnlimited 

Default branch for manual and scheduled builds

 master 

**Continue**

5. Select the **.NET Desktop with SonarCloud** template and click **Apply**.

Others

	<b>.NET Core with SonarCloud</b> Build .NET Core and ASP.NET Core applications, and analyze with SonarCloud.
	<b>.NET Desktop with SonarCloud</b> Build and run tests for .NET Desktop or Windows Classic Desktop solutions, and analyze with SonarCloud. This template requires that Visual Studio be installed on the build agent.

**Apply**

6. This build definition is pretty standard for a .NET project, except that it also includes three additional tasks for **SonarCloud**. Note that you could easily integrate these specific tasks with your existing build definitions. There's no need to start from scratch like we are in this lab.

**Pipeline**  
Build pipeline

Get sources  
PartsUnlimited master

**Agent job 1**  
Run on agent

- Use NuGet 4.4.1  
NuGet Tool Installer
- NuGet restore  
NuGet
- Prepare analysis on SonarCloud**  
Prepare Analysis Configuration
- Build solution \*\*\\*.sln  
Visual Studio Build
- VsTest - testAssemblies  
Visual Studio Test
- Run Code Analysis**  
Run Code Analysis
- ~~Publish Quality Gate Result~~  
Disabled: Publish Quality Gate Result

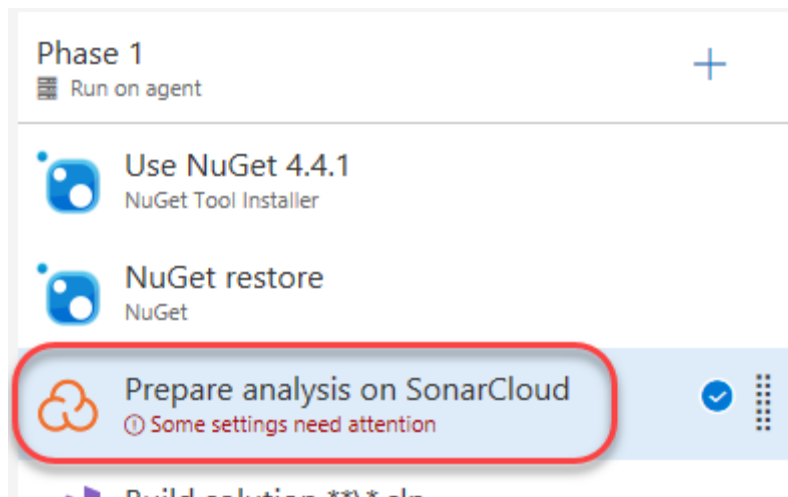
7. Set the **Agent pool** to **Hosted**. It should be the first option and have the **Visual Studio** logo.

Name \*  
Parts Unlimited Refresh-.NET Desktop with SonarCloud-CI

Agent pool \* ⓘ | [Pool information](#) | [Manage](#) ↗

**Hosted** ▼ ↻

8. Select the **Prepare analysis on SonarCloud** task. This task defines the connection configuration for any later tasks.



9. Click **New** to configure a new **SonarCloud Service Endpoint**.

Display name \*

SonarCloud Service Endpoint \* ⓘ | [Manage](#)

↕
↻
+ New

ⓘ This setting is required.

10. Click **your SonarCloud account security page** to open the account page in a new tab.

Connection name

SonarCloud Token  ⓘ

Go to your SonarCloud account security page to generate a token.

11. Sign in to your SonarCloud account.
12. To generate a token, enter a name like **"azuredevops"** and click **Generate**.

**Generate Tokens**

Generate

13. When the token is generated, click **Copy** to copy it to your clipboard. This token is tied to your account and the only thing necessary to access the service on your behalf.

New token "azuredevops" has been created. Make sure you copy it now, you won't be able to see it again!

Copy

8d94618466f010b4944baa833986c7317b6704cc

14. Close the browser tab to return to the tab with the build definition.
15. Enter a **Connection name** of "**SonarCloud**" and paste the token as the **SonarCloud Token**. Click **Verify connection** to make sure it works and click **OK**.

## Add SonarCloud endpoint

Connection name

SonarCloud

SonarCloud Token

.....



Go to your [SonarCloud account security page](#) to generate a token.

Connection: Verified

Verify connection

OK

Close

16. Select **SonarCloud** as the **SonarCloud Service Endpoint** and select an **Organization** associated with the account. You'll also need to provide a globally unique **Project Key**, such as "**partsunlimited.YOURNAME**".

Display name \*

Prepare analysis on SonarCloud

SonarCloud Service Endpoint \* ⓘ | [Manage](#)

SonarCloud



+ New

Organization \* ⓘ

.....



Choose the way to run the analysis \* ⓘ



Integrate with MSBuild



Integrate with Maven or Gradle

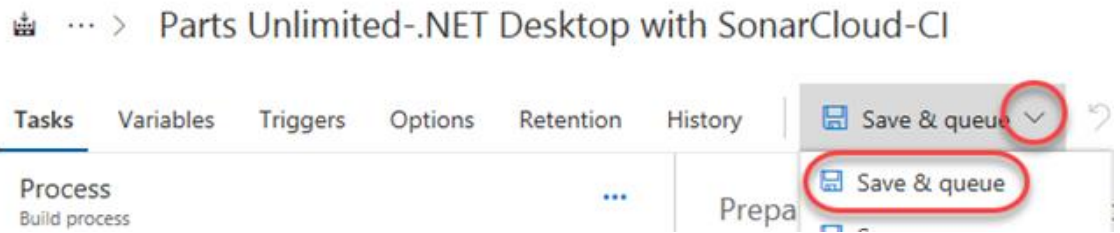


Use standalone scanner

Project Key \* ⓘ

partsunlimited.johndoe

17. Select **Save & queue** | **Save & queue** to kick off a new build.

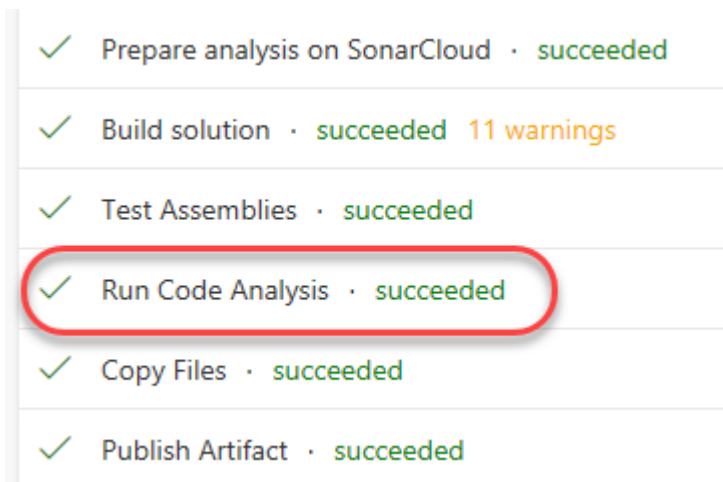


18. Click the new build link to follow its progress through to completion.

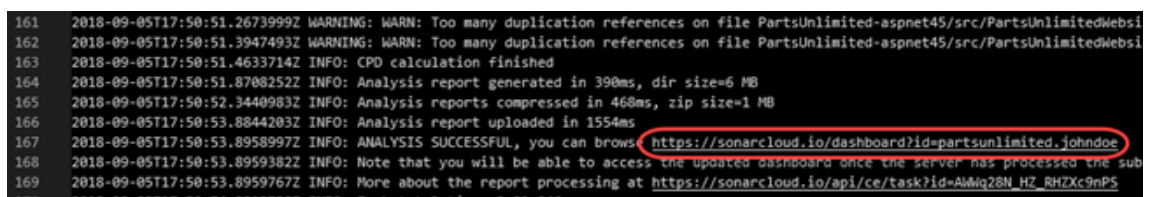


### Task 3: Reviewing SonarCloud results

1. From the left panel, select the **Run Code Analysis** task. This contains the processes where SonarCloud analyzes the code.

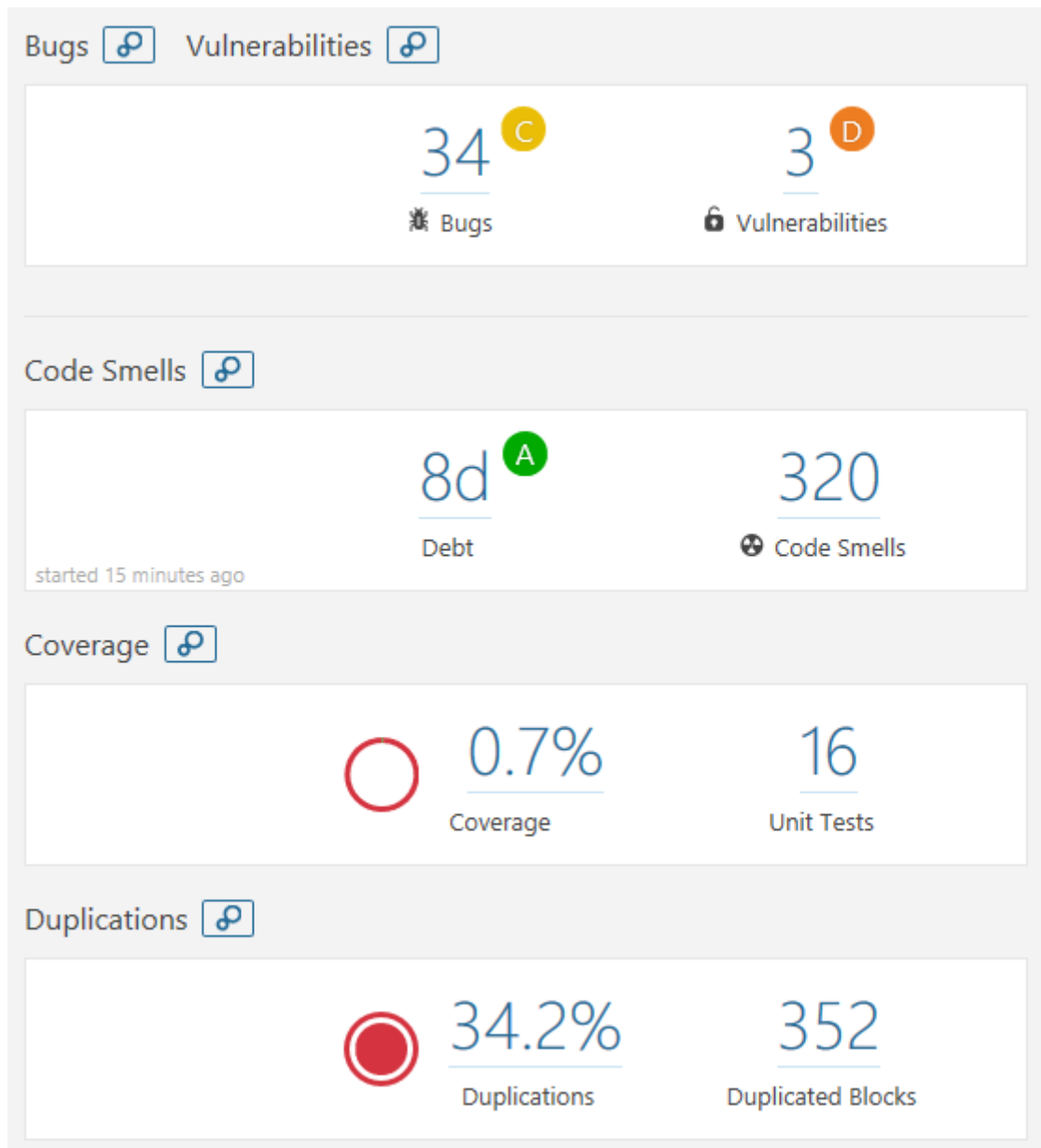


2. Near the end of the log, locate the URL to the results viewer and open it.



3. The SonarCloud results are organized for easy access to the key results you're looking for.





4. Select the **Issues** tab. This provides a convenient way to filter and sort the results so that you can attack the section you feel needs immediate attention. Select the first result.

The screenshot shows the **Issues** tab selected in the navigation bar. The left sidebar contains filters for Display Mode (Issues), Type (Bug, Vulnerability, Code Smell), and Severity (Blocker, Critical, Major, Minor, Info). The main area displays a list of issues for the file `PartsUnlimitedWebsite / App_Start/BundleConfig.cs`. The first issue is highlighted with a red box:

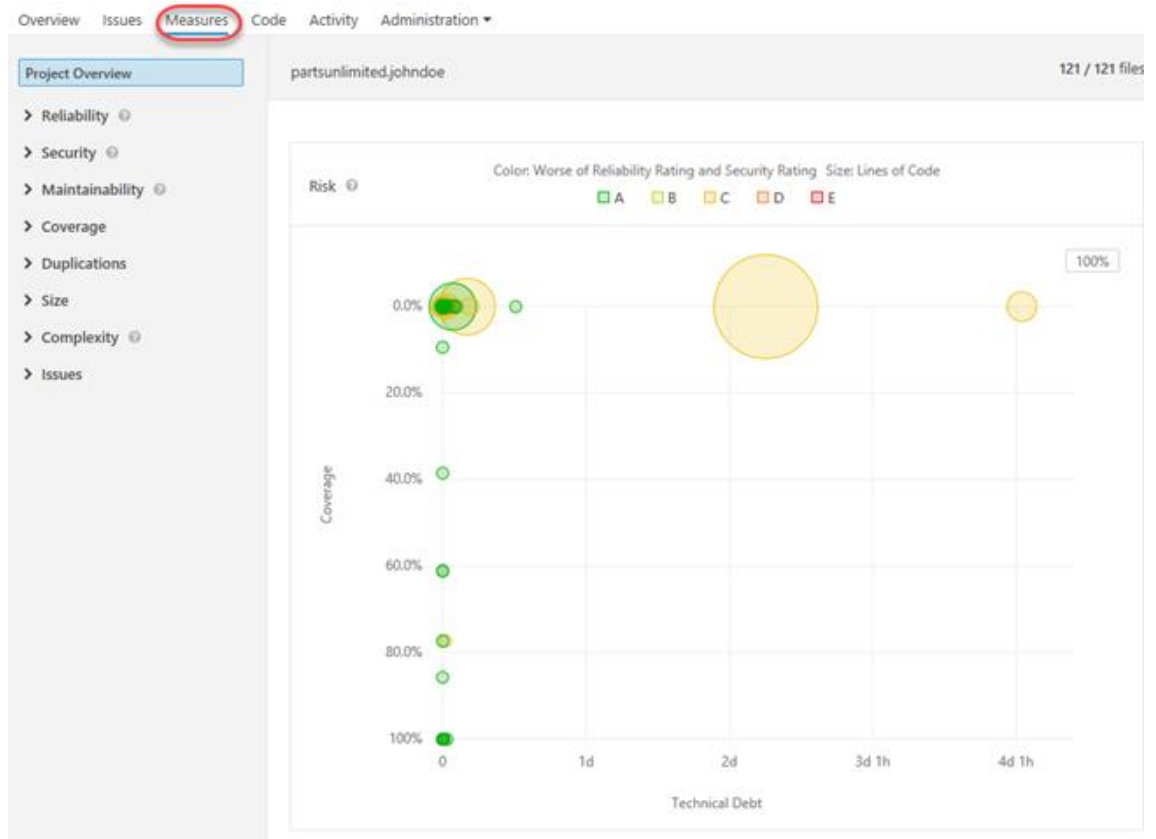
- Add a 'protected' constructor or the 'static' keyword to the class declaration.** (Code Smell, Major, Open, Not assigned, 10min effort)

Other visible issues include "Refactor your code not to use hardcoded absolute paths or URIs." (Code Smell, Minor, Open, Not assigned, 20min effort).

- The code view provides an in-depth review of each issue, along with suggestions and configuration options. For this issue, select **Open | Resolve as won't fix**.



- Select the **Measures** tab. This provides a visualization of issues as selected by the available filters.



- Filter down to see the **Reliability | Overview**. This enables you to hover over the various assets to see the amount of effort required to fix and/or maintain various components for reliability.



8. Select the **Code** tab and drill into the **PartsUnlimitedWebsite** project. This provides a way to review project issues at a file level.

Overview Issues Measures **Code** Activity Administration ▾

Search for files and sub-projects...

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Coverage	Duplications
partsunlimited.johndoe						
FabrikamFiber.SeleniumTests		0	0	0		
PartsUnlimited.UnitTests		0	0	0		
<b>PartsUnlimitedWebsite</b>	40k	34	3	320	0.7%	34.2%

9. Open **Bootstrapper.cs**.

partsunlimited.johndoe > PartsUnlimitedWebsite

Lines of Code

PartsUnlimitedWebsite	40k
<b>Bootstrapper.cs</b>	24

10. Locate the first issue related to having commented code. Expand it using the **chevron** and click the **ellipses** to see a detailed explanation and references as to the relevance and importance of this rule.



### Sections of code should not be "commented out"

Code Smell Major Main sources misra, unused Available Since 05/19/2015 SonarAnalyzer (C#) Constant/issue: 5min

Programmers should not comment out code as it bloats programs and reduces readability.

Unused code should be deleted and can be retrieved from source control history if required.

11. Expand the **Administration** option. Note that there is an incredible amount of flexibility available here for customizing your SonarCloud analysis.

